IFW AB/
2162

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

In re Application of:	§	Group Art Unit: 2162
	§	
Shanthaveeraiah, et al.	§	Examiner: Truong, Cam Y T
	§	
	§	Atty. Dkt. No.: 5181-79200 P5623
Serial No. 09/842,495	§	
Filed: April 25, 2001	§	
For: System and Method for On-Demand Node Creation for Fabric Devices	§	

CERTIFICATE OF MAILING
37 C.F.R. § 1.8

I hereby certify that this correspondence is being deposited with the U.S. Postal Service with sufficient postage as First Class Mail in an envelope addressed to: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, on the date indicated below:

Robert C. Kowert
Name of Registered Representative

June 10, 2005
Date

Signature

APPEAL BRIEF

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir/Madam:

Further to the Notice of Appeal filed April 11, 2005, Appellants present this Appeal Brief. Appellants respectfully request that the Board of Patent Appeals and Interferences consider this appeal.

I. REAL PARTY IN INTEREST

As evidenced by the assignment recorded at Reel/Frame 011772/0958, the subject application is owned by Sun Microsystems, Inc., a corporation organized and existing under and by virtue of the laws of the State of Delaware, and now having its principal place of business at 4150 Network Circle, Santa Clara, CA 95054.

II. RELATED APPEALS AND INTERFERENCES

No other appeals, interferences or judicial proceedings are known which would be related to, directly affect or be directly affected by or have a bearing on the Board's decision in this appeal.

III. STATUS OF CLAIMS

Claims 1-15, 23-59, 67 and 68 stand finally rejected. The rejection of claims 1-15, 23-59, 67 and 68 is being appealed. A copy of claims 1-15, 23-30, 32-59, 67 and 68 as currently pending is included in the Claims Appendix herein below.

IV. STATUS OF AMENDMENTS

No amendments to the claims have been submitted subsequent to the final rejection.

V. SUMMARY OF CLAIMED SUBJECT MATTER

Storage area networks, also referred to as SANs, are dedicated networks that connect one or more systems to storage devices and subsystems. Today, fibre channel is one of the leading technologies for SANs. In general, fibre channel encompasses three networking topologies: point-to-point, loop, and fabric. In a fibre channel fabric topology, the storage networks are constructed with network switches. A fabric can be composed of a single switch or multiple switches. Ports on fabric networks connect nodes to switches on low-latency, point-to-point connections.

Due to the number of devices capable of being attached to a fabric, discovering all fabric devices available to the host at host boot-up may be burdensome on a host system. Additionally, prior host system technologies for connecting to fabric networks may lack the ability to maintain the host's fabric device configuration across reboots and adjust for changes in the fabric.

Independent claim 1 is directed to a method for discovering fabric devices including receiving, from a fabric driver, a list of fabric devices available to a host system wherein the fabric driver may be a part of an operating system for the host system. For example, an administration application may query a fabric driver for a list of devices visible to a host system on which the application is executing, as described in the specification. For instance, a fabric driver may provide an API through which an administration application may make queries to obtain a list of devices. The method of claim 1 may also include receiving a request to select a subset of the fabric devices from the list. For instance, an administration application may display a list of devices through a graphical user interface or textual interface and allow a system administrator to select a subset of the fabric devices. *See, e.g.*, FIGs., 1, 3, 5, 7; page 9, line 6-page 10, line 7; page 10, line 22 – page 11, line 10; page 11, lines 12 – 21; page 12, lines 22- 26; page 14, lines 2 – 18; and page 15, lines 2-13.

Additionally, the method may include requesting the fabric driver to create an operating system device node in the host system for each of the fabric devices in the subset not already online, wherein each operating system device node provides a mechanism for access a corresponding one of the subset of fabric devices through the operating system executing on the host system. As described on pages 9-10 of the specification, onlining a subset of devices may include the creation of a node within the operating system of the host system for each device. Such a node may provide a reference for applications or other processes of in the host system to reference a corresponding device in the fabric. Thus, a node provides a path for an application or process to communicate with one of the fabric devices. *See, e.g.*, FIGs., 1, 3, 5, 7; page 8, line 29 – page 9, line 4; page 9, line 28 – page 10, line 20; page 11, lines 6-10; page 11, line 19 – page 12, line 2; page 12, lines 22- 26; page 14, lines 2 – 18; and page 15, lines 2-13.

Independent claim 43 is directed to a computer readable medium with data representing sequences of instructions executable to implement a method as described above regarding independent claim 1.

Independent claim 4 is also directed to a method for discovering fabric devices that may include providing a list of fabric devices available to a host system and receiving a request to create operating system device nodes in the host system for each fabric device in a selected subset of the fabric devices available to the host system. For example, a fabric driver may provide a list of fabric devices to an administration application in response to receiving a request, such as through an API, from the application. The method of claim 4 may also include creating an operating system device node in the host system for each of the fabric devices in the selected subset not already online. As described above regarding the method of claim 1, each operating system device node may provide a mechanism for accessing a corresponding fabric device of the selected subset through an operating system executing on the host system. *See, e.g.*, FIGs., 1, 3, 5, 7; page 8, line 29 – page 9, line 4; page 9, line 28 – page 10, line 20; page 11, lines 6-10; page 11, line 19 – page 12, line 2; page 12, lines 22- 26; page 14, lines 2 – 18; and page 15, lines 2-13.

Independent claim 47 is directed to a computer readable medium with data representing sequences of instructions executable to implement a method as described above regarding independent claim 4.

Independent claim 9 is directed toward a method for discovering devices attached to a storage network including receiving a request to identify devices attached to the storage network that are available to the host system. Additionally the method may include requesting the storage network to identify devices attached to the storage network that are available to the host system and receiving the list of identified devices. For example, as described in the specification on page 14, a fabric driver may be requested to identify only those devices available to a host system and the administration application may request the onlining of a subset of fabric devices selected either with or without a user's involvement. Thus, the method of claim 9 may also include receiving a request to on-line a subset of the identified devices and creating an operating system device node within the host system for each of the identified devices in the subset that is not already online. As with the other methods described above, each operating system device node may provide a mechanism for accessing a corresponding device of the subset of identified devices through an operating system executing on the host system. *See, e.g.*, FIGs., 1, 3, 5, 7; page 8, line 29 – page 9, line 4; page 9, line 28 – page 10, line 20; page 11, lines 6-10; page 11, line 19 – page 12, line 2; page 12, lines 22- 26; page 14, lines 2 – 18; and page 15, lines 2-13.

Independent claim 52 is directed to a computer readable medium with data representing sequences of instructions executable to implement a method as described above regarding independent claim 9.

Independent claim 23 is directed toward a host system including one or more adaptor ports for connecting to a fabric, a fabric driver and an application. The fabric driver may be configured to interface the host system to the fabric and may be part of an operating system for the host system. The application may be configured to request the fabric driver to provide a list of fabric devices attached to the fabric that are visible to the host system through one of the adaptor ports. For instance, a system administrator may select a host adaptor port and may request through an administration application that a fabric driver provide a list of fabric devices on the fabric that are available or visible through the selected adaptor port. For example, a fabric driver may provide an API through which an administration application may request a list of devices visible via the adaptor port. *See, e.g.*, FIGs. 1, 3, 9; page 10, lines 9-20; page 11, lines 12-25; and page 12, lines 22-26.

The fabric driver may be configured to provide the list of fabric devices to the application in response to the request from the application and the application may be configured to indicate to the fabric driver a selected subset of the fabric devices from the list to be brought online for access from the host system. *See, e.g.*, page 16, lines 15 – 28. Additionally, the fabric driver may be configured to online the selected subset of fabric devices so that the selected subset of fabric devices are accessible from the host system and may also create operating system device nodes within the host system for each devices of the selected subset. *See, e.g.*, FIGs., 1, 3, 7, 9; page 8, line 29 – page 9, line 4; page 9, line 28 – page 10, line 20; page 11, lines 6-10; page 11, line 12 – page 12, line 2; page 12, lines 22- 26; page 14, lines 2 – 18; and page 15, lines 2-13.

Independent claim 67 is directed to a method for discovering fabric devices including viewing a list of fabric devices available to a host system and selecting a subset of the fabric devices from the list. Additionally, the method of claim 67 includes requesting that each of the fabric devices in the subset be brought online, if not already, for use from the host system. Each fabric device that is online has a corresponding operating system device node that provides a mechanism for accessing a corresponding device of the subset of identified devices through an operating system executing on the host system. *See, e.g.*, FIGs., 1, 3, 5, 7; page 8, line 29 – page 9, line 4; page 9, line 28 – page 10, line 20; page 11, lines 6-10; page 11, line 12 – page 12, line 2; page 12, lines 22-26; page 14, lines 2 – 18; and page 15, lines 2-13.

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

Please note that for clarity and ease of discussion the grounds of rejection are not presented herein in the same order in which the Examiner presented them in the Final Office Action.

1. Claims 1-9, 12, 14, 43-52, 55, 57, 59, 67 and 68 stand finally rejected under 35 U.S.C.§ 102(e) as being anticipated by Jantz et al. (U.S. Patent 6,584,499) (hereinafter "Jantz").

2. Claims 1-9, 12, 14, 43-52, 55, 57, 59, 67 and 68 stand finally rejected under 35 U.S.C. § 103(a) as being unpatentable over Jantz in view of Nolan et al. (U.S. Patent 6,640,278) (hereinafter "Nolan").

3. Claims 10, 11, 53 and 54 stand finally rejected under 35 U.S.C. § 103(a) as being unpatentable over Jantz in view of Blumenau et al. (U.S. Patent 6,665,714) (hereinafter "Blumenau").

4. Claims 10, 11, 53 and 54 stand finally rejected under 35 U.S.C. § 103(a) as being unpatentable over Jantz in view of Nolan and further in view of Blumenau.

5. claims 13 and 56 stand finally rejected under 35 U.S.C. § 103(a) as being unpatentable over Jantz in view of Carlson et al. (U.S. Patent 5,600,791) (hereinafter "Carlson") and Basham.

6. Claims 13 and 56 stand finally rejected under 35 U.S.C. § 103(a) as being unpatentable over Jantz in view of Nolan and further in view of Carlson and Basham et al. (U.S. Patent 6,182,167) (hereinafter "Basham").

7. Claims 15 and 58 stand finally rejected under 35 U.S.C. § 103(a) as being unpatentable over Jantz in view of Wieland (U.S. Patent 6,643,748).

8. Claims 15 and 58 stand finally rejected under 35 U.S.C. § 103(a) as being unpatentable over Jantz in view of Nolan and further in view of Wieland.

9. Claims 23-25, 28-30 and 40-42 stand finally rejected under 35 U.S.C. § 103(a) as being unpatentable over Blumenau in view of Nolan.

10. Claim 26 stands finally rejected under 35 U.S.C. § 103(a) as being unpatentable over Blumenau in view of Nolan and further in view of Nolan (U.S. Patent 6,466,141) (hereinafter "Nolan '141").

11. Claim 27 stands finally rejected under 35 U.S.C. § 103(a) as being unpatentable over Blumenau in view of Nolan and further in view of Nolan '141 and Sambamurthy et al. (U.S. Patent 6,393,489) (hereinafter "Sambamurthy").

12. Claim 31 stands finally rejected under 35 U.S.C. § 103(a) as being unpatentable over Blumenau in view of Nolan and further in view of Jantz.

13. Claims 32-37 stand finally rejected under 35 U.S.C. § 103(a) as being unpatentable over Blumenau in view of Nolan and further in view of Chow et al. (U.S. Patent 6,594,698) (hereinafter "Chow").

14. Claims 38-39 stand finally rejected under 35 U.S.C. § 103(a) as being unpatentable over Blumenau in view of Nolan and further in view of Chow, Carlson and Basham.

VII. ARGUMENT

As noted above, the grounds of rejection are not presented in the same order in which the Examiner presented them in the Final Office Action. For clarity, however, the details of each ground of rejection are repeated before arguments are presented.

First Ground of Rejection:

Claims 1-9, 12, 14, 43-52, 55, 57, 59, 67 and 68 stand finally rejected under 35 U.S.C.§ 102(e) as being anticipated by Jantz et al. (U.S. Patent 6,584,499) (hereinafter "Jantz"). Appellants traverse this rejection for at least the following reasons. Different groups of claims are addressed under their respective subheadings.

Claims 1 and 2:

Regarding claim 1, Jantz does not disclose requesting a fabric driver that is part of an operating system for the host system to create an operating system device node in the host system for each of a subset of fabric devices not already online, wherein each operating system device node provides a mechanism for accessing a corresponding one of the subset of fabric devices through the operating system executing on the host system.

Instead, Jantz teaches a method for configuring and monitoring manageable network devices. The device configuration and monitoring described in Jantz *has absolutely nothing to do with* how the devices are accessed through a host system's operating system. The Examiner refers to the description of Jantz's discover-monitor application (DMA) at col. 14, lines 1-30. However, Jantz's discover-monitor application is not a fabric driver that is part of an operating system for the host system. In fact, Jantz describes very clearly that his DMA executes within a browser application (*see*, Jantz, column 15, lines 5-43). For example Jantz states, "[d]uring the browser start-up process, the user preferably supplies the URL of the processs-discover-monitor applet to run" (Jantz, column 15, lines 7-9). Thus, Jantz's discover-monitor application and other components are *Java applets* (which are clearly not fabric drivers) functioning at an application layer and not part of an operating system.

Additionally, Jantz also fails to disclose receiving a list from a fabric driver of fabric devices available to a host system, wherein the fabric driver is part of an operating system for the host system. Jantz's discover-monitor application does not receive a list of fabric devices available to a host system from a fabric driver that is part of an operating system for the host system. Instead, Jantz describes two methods for obtaining a list of devices. In the first method, a user "preferably enters the device into DMA 822, and DMA 822 then starts a monitor thread 824 for the entered device" (Jantz, column 15, lines 44-51). In other words, a user of Jantz system may manually input information about a device. Alternatively, Jantz's discover-monitor application may automatically discover all direct network attached devices and all servers. The discover-monitor application then obtains from each server a list of devices it knows about. Jantz does not describe receiving a list *from a fabric driver* of fabric devices available to a host system, wherein the fabric driver is part of an operating system for the host system.

Furthermore, Jantz does not disclose requesting the fabric driver to create an operating system device node in the host system for each of the fabric devices in the subset not already online, wherein each operating system device node provides a mechanism for accessing a corresponding one of the subset of fabric devices through the operating system executing on the host system. The Examiner refers to the device connection table described at col. 16, lines 13-20 of Jantz. However, Jantz teaches that the device connection table is simply a connection map created from information obtained from the device controller 806 and thus *is clearly not a system device node* in a host system that provides a mechanism for accessing the device through the operating system on the host system. Instead, Jantz teaches that the connection table is used by the discover-monitor applet 822 to display the device connections, as is shown in Figs. 6 and 7 of Jantz. The connection table in Jantz clearly *has nothing to do with* operating system device nodes. Jantz's teachings pertain to an application monitoring and configuring of devices on a network. Jantz's teachings have no relevance to creating an operating system device node in the host system for each of a subset of fabric devices not already online, as recited in appellant's claim 1.

In response to Appellants' argument above, the Examiner, in the Advisory Action of March 28, 2005, equates the building of a device connection table by DMA 822 in Jantz with creating an operating system device node. However, a connection table providing a list of connections into each of a number of devices is clearly not an operating system device node. Operating system device nodes are well understood in the art and one skilled in the art would not consider Jantz' connection table as an operating system device node. Additionally, as noted above, the connection table is part of a JAVA applet and not part of an operating system. Furthermore Jantz' connection table does not provide a mechanism for accessing a corresponding fabric device *through the operating system*. Instead, Jantz' connection table is only available to Jantz' discover-monitor application (DMA 822). The Examiner also states in the Advisory Action that Jantz's building of a connection table "implies that a device connection table is not already online." This assertion by the Examiner has no relevance to Appellants' arguments or to the claim limitations.

Furthermore, the Examiner admits in the § 103 rejection of claim 1 that Jantz fails to disclose "wherein the fabric driver is part of an operating system for the host system; requesting the fabric driver to create an operating system device node in the host system for each of the fabric devices in the subset not already online" (see, Final Office Action, page 10, paragraph 3). **Thus, by the Examiner's own admission, Jantz does not anticipate Appellants' claim 1.**

Claim 3:

Regarding claim 3, Jantz fails to disclose requesting the fabric driver to provide the list of fabric devices available to the host system in response to user input. The Examiner cites FIG. 6 and column 13, lines 15-50 of Jantz describing the discover-monitor application displaying a list of devices. However, merely displaying a list of devices does not imply requesting such a list from a fabric driver. As noted above regarding claim 1, Jantz' discover-monitor application does not request a fabric driver to provide a list of fabric devices. Instead, Jantz describes two methods for obtaining a list of devices. In the first method, a user "preferably enters the device into DMA 822, and DMA 822 then starts a monitor thread 824 for the entered device" (Jantz, column 15, lines 44-51). In other words, a user of Jantz system may manually input information about a device. Alternatively, Jantz's discover-monitor application may automatically discover all direct network attached devices and all servers. The discover-monitor application then obtains from each server a list of devices it knows about. Nowhere does Jantz describe requesting the fabric driver to provide a list of fabric devices available to the host system in response to user input.

Claims 4, 5 and 8:

Regarding claim 4, Jantz fails to disclose receiving a request to create operating system device nodes in the host system for each fabric device in a selected subset of the fabric device available to the host system. The Examiner refers to Jantz management protocol server being queried via an RPC agent thread for its associated device properties, citing column 16, lines 13-20 of Jantz. However, a query for device properties is very different from a request to create operating system nodes in a host system.

Jantz also fails to disclose creating an operating system device node in the host system for each of the fabric devices in the selected subset not already online, wherein each operating system device node provides a mechanism for accessing a corresponding one of the subset of fabric devices through an operating system executing on the host system. The Examiner refers to the device connection table described at col. 16, lines 13-20 of Jantz. However, as noted above regarding claim 1, Jantz teaches that the device connection table is simply a connection map created from information obtained from the device controller 806 and thus is clearly not a system device node in a host system that provides a mechanism for accessing the device through the operating system on the host system.

Instead, Jantz teaches that the connection table is used by the discover-monitor applet 822 to display the device connections, as is shown in Figs. 6 and 7 of Jantz. The connection table in Jantz clearly has nothing to do with operating system device nodes. Jantz's teachings pertain to the monitoring and configuring of devices on a network and have nothing to do with creating an operating system device node in the host system for each of a subset of fabric devices not already online. Furthermore Jantz's connection table does not provide any mechanism for accessing fabric devices through the operating system executing on the host system. Instead, as noted above, Jantz's connection table merely stores connection map for devices.

Jantz contains no teachings at all in regard to providing operating system device nodes that provide a mechanism for accessing corresponding devices through an operating system executing on the host system, as discussed above.

Furthermore, the Examiner admits in the § 103 rejection of claim 4 that Jantz fails to disclose requesting the fabric driver to create an operating system device node in the host system for each of the fabric devices in the subset not already online. **Thus, by the Examiner's own admission, Jantz does not anticipate Appellants' claim 4.**

Claims 6 and 7:

Regarding claim 6, Jantz fails to disclose wherein said compiling the list of fabric devices available to the host system comprises: from the information about the fabric devices, selecting the fabric devices supporting one protocol out of a plurality of protocols supported on the fabric and compiling the list of fabric devices to list only those fabric devices supporting the one protocol. The Examiner cites FIG. 8 and column 15, line 50 through column 16, line 20 of Jantz. However, the cited portions refer to discovering "all the devices on the network" (Jantz, column 15, line 52-55). The cited portions of Jantz do not mention selecting fabric devices that support one protocol out of a plurality of protocols supported on the fabric.

Claims 9 and 12:

Regarding claim 9, Jantz fails to disclose receiving a request to on-line a subset of the identified devices. The Examiner cites column 13, lines 45-55 and refers to a user selecting a device node in a user interface causing the device's associated management interface application program to be launched. However, launching a management application is clearly not the same as receiving a request to on-line a subset of the identified devices. Furthermore, Jantz does not teach that launching a device's management interface application program involves either requesting or receiving a request to on-line a device.

Jantz further fails to disclose creating an operating system device node within the host system for each of the identified devices in the subset that is not already online, wherein each operating system device node provides a mechanism for accessing a corresponding one of the subset of the identified devices through an operating system executing on the host system. The Examiner cites column 16, lines 13-20 where Jantz describes building a device connection table that provides, for each device, a list of connections into the device. As noted above, connection table is simply a connection map created from information obtained from the device controller 806 and thus is clearly not a system device node in a host system. For a more detailed discussion regarding Jantz connection table, please refer to the remarks above regarding claims 1 and 4.

Appellants again note that Jantz contains absolutely no teachings at all in regard to providing operating system device nodes that provide a mechanism for accessing corresponding devices through an operating system executing on the host system, as discussed above.

Furthermore, the Examiner admits in the § 103 rejection of claim 9 that Jantz fails to disclose requesting the fabric driver to create an operating system device node in the host system for each of the fabric devices in the subset not already online. **Thus, by the Examiner's own admission, Jantz does not anticipate Appellants' claim 9.**

Claim 14:

Jantz fails to disclose in response to a reboot of the host system: reading the persistent repository; and onlining the devices indicated by the persistent repository to have been onlined prior to the reboot. The Examiner cites column 14, liens 14-45 of Jantz. However, the cited portion of Jantz fails to mention a reboot of a host system, reading a persistent repository, or onlining devices indicated by the persistent repository to have been onlined prior to the reboot. Instead, the cited passage describes how a user may select and load a management interface application program for a device by selecting the device from the discover-monitor application screen. Nowhere does Jantz mention onlining devices after a reboot based upon those devices indicated by a persistent repository to have been onlined prior to the reboot.

Claims 43, 44 and 45:

Regarding claim 43, Jantz does not disclose requesting a fabric driver that is part of an operating system for the host system to create an operating system device node in the host system for each of a subset of fabric devices not already online, wherein each operating system device node provides a mechanism for accessing a corresponding one of the subset of fabric devices through the operating system executing on the host system.

Instead, Jantz teaches a method for configuring and monitoring manageable network devices. The device configuration and monitoring described in Jantz *has absolutely nothing to do with* how the devices are accessed through a host system's operating system. The Examiner refers to the description of Jantz's discover-monitor application (DMA) at col. 14, lines 1-30. However, Jantz's discover-monitor application is not a fabric driver that is part of an operating system for the host system. In fact, Jantz describes very clearly that his DMA executes within a browser application (see, Jantz, column 15, lines 5-43). For example Jantz states, "[d]uring the browser start-up process, the user preferably supplies the URL of the processs-discover-monitor applet to run" (Jantz, column 15, lines 7-9). Thus, Jantz's discover-monitor application and other components are *Java applets* (which are clearly not fabric drivers) functioning at an application layer and not part of an operating system.

Additionally, Jantz also fails to disclose receiving a list from a fabric driver of fabric devices available to a host system, wherein the fabric driver is part of an operating system for the host system.

Jantz's discover-monitor application does not receive a list of fabric devices available to a host system from a fabric driver that is part of an operating system for the host system. Instead, Jantz describes two methods for obtaining a list of devices. In the first method, a user "preferably enters the device into DMA 822, and DMA 822 then starts a monitor thread 824 for the entered device" (Jantz, column 15, lines 44-51). In other words, a user of Jantz system may manually input information about a device. Alternatively, Jantz's discover-monitor application may automatically discover all direct network attached devices and all servers. The discover-monitor application then obtains from each server a list of devices it knows about. Jantz does not describe receiving a list *from a fabric driver* of fabric devices available to a host system, wherein the fabric driver is part of an operating system for the host system.

Furthermore, Jantz does not disclose requesting the fabric driver to create an operating system device node in the host system for each of the fabric devices in the subset not already online, wherein each operating system device node provides a mechanism for accessing a corresponding one of the subset of fabric devices through the operating system executing on the host system. The Examiner refers to the device connection table described at col. 16, lines 13-20 of Jantz. However, Jantz teaches that the device connection table is simply a connection map created from information obtained from the device controller 806 and thus *is clearly not a system device node* in a host system that provides a mechanism for accessing the device through the operating system on the host system. Instead, Jantz teaches that the connection table is used by the discover-monitor applet 822 to display the device connections, as is shown in Figs. 6 and 7 of Jantz. The connection table in Jantz clearly *has nothing to do with* operating system device nodes. Jantz's teachings pertain to an application monitoring and configuring of devices on a network. Jantz's teachings have no relevance to creating an operating system device node in the host system for each of a subset of fabric devices not already online, as recited in appellants claim 1.

In response to Appellants' argument above, the Examiner, in the Advisory Action, equates the building of a device connection table by DMA 822 in Jantz with creating an operating system device node. However, a connection table providing a list of connections into each of a number of devices is clearly not an operating system device node. Operating system device nodes are well understood in the art and one skilled in the art would not consider Jantz' connection table as an operating system device node. Additionally, as noted above, the connection table is part of a JAVA applet and not part of an operating system. Furthermore Jantz' connection table does not mechanism for accessing a corresponding fabric device *through the operating system*. Instead, Jantz' connection table is only available to Jantz' discover-monitor application (DMA 822). The Examiner also states in the Advisory action that Jantz' building of a connection table "implies that a device connection table is not already online." This assertion by the Examiner has no relevance to Appellants' arguments.

Furthermore, the Examiner admits in the § 103 rejection of claim 43 that Jantz fails to disclose "wherein the fabric driver is part of an operating system for the host system; requesting the fabric driver to create an operating system device node in the host system for each of the fabric devices in the subset not already online" (see, Final Office Action, page 10, paragraph 3). **Thus, by the Examiner's own admission, Jantz does not anticipate Appellants' claim 43.**

Claim 46:

Regarding claim 46, Jantz fails to disclose requesting the fabric driver to provide the list of fabric devices available to the host system in response to user input. The Examiner cites FIG. 6 and column 13, lines 15-50 of Jantz describing the discover-monitor application displaying a list of devices. However, merely displaying a list of devices does not imply requesting such a list from a fabric driver. As noted above regarding claim 1, Jantz' discover-monitor application does not request a fabric driver to provide a list of fabric devices. Instead, Jantz describes two methods for obtaining a list of devices. In the first method, a user "preferably enters the device into DMA 822, and DMA 822 then starts a monitor thread 824 for the entered device" (Jantz, column 15, lines 44-51). In other words, a user of Jantz system may manually input information about a device. Alternatively, Jantz's discover-monitor application may automatically discover all direct network attached devices and all servers. The discover-monitor application then obtains from each server a list of devices it knows about. Nowhere does

Jantz describe requesting the fabric driver to provide a list of fabric devices available to the host system in response to user input.

Claims 47, 48 and 51:

Regarding claim 47, Jantz fails to disclose receiving a request to create operating system device nodes in the host system for each fabric device in a selected subset of the fabric device available to the host system. The Examiner refers to Jantz management protocol server being queried via an RPC agent thread for its associated device properties, citing column 16, lines 13-20 of Jantz. However, a query for device properties is very different from a request to create operating system nodes in a host system.

Jantz also fails to disclose creating an operating system device node in the host system for each of the fabric devices in the selected subset not already online, wherein each operating system device node provides a mechanism for accessing a corresponding one of the subset of fabric devices through an operating system executing on the host system. The Examiner refers to the device connection table described at col. 16, lines 13-20 of Jantz. However, as noted above regarding claim 1, Jantz teaches that the device connection table is simply a connection map created from information obtained from the device controller 806 and thus is clearly not a system device node in a host system that provides a mechanism for accessing the device through the operating system on the host system.

Instead, Jantz teaches that the connection table is used by the discover-monitor applet 822 to display the device connections, as is shown in Figs. 6 and 7 of Jantz. The connection table in Jantz clearly has nothing to do with operating system device nodes. Jantz's teachings pertain to the monitoring and configuring of devices on a network and have nothing to do with creating an operating system device node in the host system for each of a subset of fabric devices not already online. Furthermore Jantz's connection table does not provide any mechanism for accessing fabric devices through the operating system executing on the host system. Instead, as noted above, Jantz's connection table merely stores connection map for devices.

Please note that Jantz contains no teachings at all in regard to providing operating system device nodes that provide a mechanism for accessing corresponding devices through an operating system executing on the host system, as discussed above.

Furthermore, the Examiner admits in the § 103 rejection of claim 47 that Jantz fails to disclose requesting the fabric driver to create an operating system device node in the host system for each of the fabric devices in the subset not already online. **Thus, by the Examiner's own admission, Jantz does not anticipate Appellants' claim 47.**

Claims 49 and 50:

Regarding claim 49, Jantz fails to disclose wherein said compiling the list of fabric devices available to the host system comprises: from the information about the fabric devices, selecting the fabric devices supporting one protocol out of a plurality of protocols supported on the fabric and compiling the list of fabric devices to list only those fabric devices supporting the one protocol. The Examiner cites FIG. 8 and column 15, line 50 through column 16, line 20 of Jantz. However, the cited portions refer to discovering "all the devices on the network" (Jantz, column 15, line 52-55). The cited portions of Jantz do not mention selecting fabric devices that support one protocol out of a plurality of protocols supported on the fabric.

Claims 52, 55 and 59:

Regarding claim 52, Jantz fails to disclose receiving a request to on-line a subset of the identified devices. The Examiner cites column 13, lines 45-55 and refers to a user selecting a device node in a user interface causing the device's associated management interface application program to be launched. However, launching a management application is clearly not the same as receiving a request to on-line a subset of the identified devices. Furthermore, Jantz does not teach that launching a device's management interface application program involves either requesting or receiving a request to on-line a device.

Jantz further fails to disclose creating an operating system device node within the host system for each of the identified devices in the subset that is not already online, wherein each operating system device node provides a mechanism for accessing a corresponding one of the subset of the identified devices through an operating system executing on the host system. The Examiner cites column 16, lines 13-20 where Jantz describes building a device connection table that provides, for each device, a list of connections into the device. As noted above, connection table is simply a connection map created from information obtained from the device controller 806 and thus is clearly not a system device node in a host system. For a more detailed discussion regarding Jantz connection table, please refer to the remarks above regarding claims 1 and 4.

Appellants again note that Jantz contains absolutely no teachings at all in regard to providing operating system device nodes that provide a mechanism for accessing corresponding devices through an operating system executing on the host system, as discussed above.

Furthermore, the Examiner admits in the § 103 rejection of claim 52 that Jantz fails to disclose requesting the fabric driver to create an operating system device node in the host system for each of the fabric devices in the subset not already online. **Thus, by the Examiner's own admission, Jantz does not anticipate Appellants' claim 52.**

Claim 57:

Jantz fails to disclose in response to a reboot of the host system: reading the persistent repository; and onlining the devices indicated by the persistent repository to have been onlined prior to the reboot. The Examiner cites column 14, liens 14-45 of Jantz. However, the cited portion of Jantz fails to mention a reboot of a host system, reading a persistent repository, or onlining devices indicated by the persistent repository to have been onlined prior to the reboot. Instead, the cited passage describes how a user may select and load a management interface application program for a device by selecting the device from the discover-monitor application screen. Nowhere does Jantz mention onlining devices after a reboot based upon those devices indicated by a persistent repository to have been onlined prior to the reboot.

Claims 67 and 68:

Regarding claim 67, Examiner admits that Jantz does not disclose requesting that each of the fabric devices in the subset be brought online if not already online for use from the host system. The Examiner cites column 16, lines 10-25 and refers to Jantz's teachings regarding building a connection table and Jantz teachings regarding displaying information regarding discovered devices on a display screen. The Examiner argues that Jantz's building of a connection table and displaying all discovered devices indicates that Jantz's system "brings all discovered devices online". The Examiner is incorrect.

Building a connection table including connection to device mappings and that indicates which management application program is associated with each device does not imply that devices must have been brought online. The ability to display and manipulate information about devices does not require or imply bring those devices online. For instance, information about devices, connections and management programs can be manipulated (i.e. mappings built and information displayed) regardless of whether the devices are online. Secondly, Jantz does not mention anything about bringing devices online as part of either building his connection table or displaying discovered devices. Thus, the Examiner's rejection is clearly based on hindsight analysis and speculation.

Furthermore, the Examiner admits in the § 103 rejection of claim 67 that Jantz fails to disclose requesting the fabric driver to create an operating system device node in the host system for each of the fabric devices in the subset not already online. **Thus, by the Examiner's own admission, Jantz does not anticipate Appellants' claim 67.**

Second Ground of Rejection:

Claims 1-9, 12, 14, 43-52, 55, 57, 59, 67 and 68 stand finally rejected under 35 U.S.C. § 103(a) as being unpatentable over Jantz in view of Nolan et al. (U.S. Patent 6,640,278) (hereinafter "Nolan"). Appellants traverse this rejection for at least the following reasons. Different groups of claims are addressed under their respective subheadings.

Claim 1:

Regarding claim 1, Jantz in view of Nolan does not teach or suggest requesting a fabric driver that is part of an operating system for the host system to create an operating system device node in the host system for each of a subset of fabric devices not already online, wherein each operating system device node provides a mechanism for accessing a corresponding one of the subset of fabric devices through the operating system executing on the host system. Instead, Jantz teaches a method for configuring and monitoring manageable network devices. The device configuration and monitoring described in Jantz *has absolutely nothing to do with* how the devices are accessed through a host system's operating system.

The Examiner refers to the description of Jantz's discover-monitor application (DMA) at col. 14, lines 1-30. However, Jantz's discover-monitor application is not a fabric driver that is part of an operating system for the host system. Instead, Jantz's discover-monitor application and other components are *Java applets* (which are clearly not fabric drivers) functioning at an application layer and hence, are not part of an operating system. In fact, Jantz describes very clearly that his DMA executes within a browser application (see, Jantz, column 15, lines 5-43).

Additionally, Jantz in view of Nolan also fails to teach or suggest receiving a list from a fabric driver of fabric devices available to a host system, wherein the fabric driver is part of an operating system for the host system. Jantz's discover-monitor application does not receive a list of fabric devices available to a host system *from a fabric driver* *that is part of an operating system* for the host system. For a more detailed discussion regarding Jantz's methods for obtaining lists of devices, please see the discussion above regarding the § 102 rejection of claim 1.

Furthermore, Jantz in view of Nolan does not teach or suggest requesting the fabric driver to create an operating system device node in the host system for each of the fabric devices in the subset not already online, wherein each operating system device node provides a mechanism for accessing a corresponding one of the subset of fabric devices through the operating system executing on the host system. The Examiner refers to the device connection table described at col. 16, lines 13-20 of Jantz. However, Jantz teaches that the device connection table is simply a connection map created from information obtained from the device controller 806 and thus *is clearly not a system device node* in a host system that provides a mechanism for accessing the device through the operating system on the host system. For a more detailed discussion regarding Jantz's connection table, please see the discussion above regarding the § 102 rejection of claim 1.

The Examiner admits that Jantz fails to disclose wherein the fabric driver is part of an operating system for the host system, and requesting the fabric driver to create an operating system device node in the host system for each of the fabric devices in the subset not already online. The Examiner states, "Nolan teaches that a server includes resources in an operating system" and argues, "[e]ach resource is represented as the fabric driver ... [t]he server also builds LUNs that are displayed to a user." The Examiner is apparently arguing that since Nolan discloses a server that includes some resources in an operating system, it would be obvious to modify Jantz to include the specific fabric driver from Appellants' claim 1. This reasoning makes no sense. The Examiner is referring to a portion of Nolan that describes the management of emulated storage devices. Nolan does not mention anything regarding a fabric driver in an operating system that is requested to create operating system device nodes in a host system. Furthermore, the Examiner's proposed combination of Jantz and Nolan would not result in a system that includes such a fabric driver (as recited in Appellants' claim 1). Instead the combination of Jantz and Nolan would result in a system that discovers and displays information regarding network devices as taught by Jantz, but that could also display information about emulated devices as in Nolan.

Claim 4:

Regarding claim 4, Jantz fails to disclose receiving a request to create operating system device nodes in the host system for each fabric device in a selected subset of the fabric device available to the host system. Jantz also fails to disclose creating an operating system device node in the host system for each of the fabric devices in the selected subset not already online, wherein each operating system device node provides a mechanism for accessing a corresponding one of the subset of fabric devices through an operating system executing on the host system.

The Examiner refers to the device connection table described at col. 16, lines 13-20 of Jantz. However, as noted above regarding claim 1, Jantz teaches that the device connection table is simply a connection map created from information obtained from the device controller 806 and thus is clearly not a system device node in a host system that provides a mechanism for accessing the device through the operating system on the host system. For a more detailed discussion regarding Jantz's connection table, please see the discussion above regarding the § 102 rejection of claim 4.

The Examiner admits that Jantz fails to disclose wherein the fabric driver is part of an operating system for the host system; and requesting the fabric driver to create an operating system device node in the host system for each of the fabric devices in the subset not already online. The Examiner argues, "Nolan teaches that a server includes resources in an operating system" and argues, "[e]ach resource is represented as the fabric driver ... [t]he server also builds LUNs that are displayed to a user." The Examiner is apparently arguing that since Nolan discloses a server that includes some resources in an operating system, it would be obvious to modify Jantz to include the specific fabric driver from Appellants' claim 4. There is clearly no such suggestion in Nolan. The

Examiner is referring to a portion of Nolan that describes the management of emulated storage devices. Nolan does not mention anything regarding a fabric driver in an operating system that is requested to create operating system device nodes in a host system. Furthermore, the Examiner's proposed combination of Jantz and Nolan would not result in a system that includes such a fabric driver (as recited in Appellants' claim 4). Instead the combination of Jantz and Nolan would result in a system that discovers and displays information regarding network devices as taught by Jantz, but that could also display information about emulated devices as in Nolan.

Claim 9:

Regarding claim 9, Jantz fails to disclose creating an operating system device node within the host system for each of the identified devices in the subset that is not already online, wherein each operating system device node provides a mechanism for accessing a corresponding one of the subset of the identified devices through an operating system executing on the host system. The Examiner cites column 16, lines 13-20 where Jantz describes building a device connection table that provides, for each device, a list of connections into the device. As noted above, connection table is simply a connection map created from information obtained from the device controller 806 and thus is clearly not a system device node in a host system. For a more detailed discussion regarding Jantz connection table, please refer to the remarks above regarding claim 1.

The Examiner admits that Jantz fails to disclose wherein the fabric driver is part of an operating system for the host system; and requesting the fabric driver to create an operating system device node in the host system for each of the fabric devices in the subset not already online. The Examiner states, "Nolan teaches that a server includes resources in an operating system" and argues, "[e]ach resource is represented as the fabric driver ... [t]he server also builds LUNs that are displayed to a user." The Examiner is apparently arguing that since Nolan discloses a server that includes some resources in an operating system, it would be obvious to modify Jantz to include the specific fabric driver from Appellants' claim 9. Nolan contains no such suggestion. The Examiner is referring to a portion of Nolan that describes the management of emulated storage devices. Nolan does not mention anything regarding a fabric driver in an operating system that is requested to create operating system device nodes in a host system. Furthermore, the Examiner's proposed combination of Jantz and Nolan would not result in a system that includes such a fabric driver (as recited in Appellants' claim 9). Instead the combination of Jantz and Nolan would result in a system that discovers and displays information regarding network devices as taught by Jantz, but that could also display information about emulated devices as in Nolan.

Claim 67:

Regarding claim 67, Examiner admits that Jantz does not disclose requesting that each of the fabric devices in the subset be brought online if not already online for use from the host system. The Examiner cites column 16, lines 10-25 and contends that Jantz's teachings regarding building a connection table, which includes a list of connections to a device and information about a which management application program is associated with a device, and Jantz teachings regarding displaying information regarding discovered devices on a display screen indicate that Jantz system brings all discovered devices online. Appellants strongly disagree. Discovering and displaying information regarding devices on a network is very different from requesting that those devices be brought online. Jantz makes no mention whatsoever regarding requesting that devices be brought online. For a more detailed discussion regarding this argument, please see the discussion above regarding the § 102 rejection of claim 67.

The Examiner admits that Jantz fails to disclose wherein the fabric driver is part of an operating system for the host system, and requesting the fabric driver to create an operating system device node in the host system for each of the fabric devices in the subset not already online. The Examiner states, "Nolan teaches that a server includes resources in an operating system" and argues, "[e]ach resource is represented as the fabric driver ... [t]he server also builds LUNs that are displayed to a user." The Examiner is apparently arguing that since Nolan discloses a server that includes some resources in an operating system, it would be obvious to modify Jantz to include the specific fabric driver from Appellants' claim 67. Nolan contains no such suggestion. The Examiner is referring to a portion of Nolan that describes the management of emulated storage devices. Nolan does not mention anything regarding a fabric driver in an operating system that is requested to create operating system device nodes in a host system. Furthermore, the Examiner's proposed combination of Jantz and Nolan would not result in a system that includes such a fabric driver (as recited in Appellants' claim 67). Instead the combination of Jantz and Nolan would result in a system that discovers and displays information regarding network devices as taught by Jantz, but that could also display information about emulated devices as in Nolan.

Third Ground of Rejection:

Claims 10, 11, 53 and 54 stand finally rejected under 35 U.S.C. § 103(a) as being unpatentable over Jantz in view of Blumenau et al. (U.S. Patent 6,665,714) (hereinafter "Blumenau"). Appellants traverse this rejection for at least the following reasons. Different groups of claims are addressed under their respective subheadings.

Claim 10:

Regarding claim 10, Jantz in view of Blumenau fails to teach or suggest wherein said request to identify devices attached to the storage network which are available to a host system is for devices available to the host system through a specified one of the ports, and wherein said requesting the storage network to identify devices is made for the specified port. The Examiner admits that Jantz does not teach or suggest requesting a storage network to identify devices available to the host system through a specified one of a plurality of ports of the host system to the fabric. The Examiner cites column 7, lines 1-10 of Blumenau and argues that Blumenau "teaches identifying the device and the port of the device that coupled to the network." However, claim 10 recites identifying devices attached to the storage network that are available to a host system through a specified port of the host system. Blumenau only teaches identifying each port of a device that couples to the network, but fails to teach anything regarding identifying those devices available through a specified port *of the host system*. Since neither Jantz nor Blumenau teaches identifying devices available through a specified port of the host system, the Examiner's proposed combination of Jantz and Blumenau would also fail to include identifying devices available through a specified port of the host system.

Claim 11:

Regarding claim 11, Jantz in view of Blumenau fails to teach or suggest wherein said request to identify devices attached to the storage network which are available to a host system is for devices available to the host system through a specified set of the ports, and wherein said requesting the storage network to identify devices is made for the specified set of the ports. The Examiner admits that Jantz does not teach or suggest requesting a storage network to identify devices available to the host system through a specified set of a plurality of ports of the host system to the fabric. The Examiner cites column 7, lines 1-10 of Blumenau and argues that Blumenau "teaches identifying the device and the port of the device that coupled to the network." However, claim 11 recites identifying devices attached to the storage network that are available to a host system through a specified set of ports *of the host system*. Blumenau only teaches identifying each port of a device that couples to the network, but fails to teach anything regarding identifying those devices available through a specified port of the host system. Since neither Jantz nor Blumenau teaches identifying devices available through a specified set of ports of the host system, the Examiner's proposed combination of Jantz and Blumenau would also fail to include identifying devices available through a specified set of ports of the host system.

Claim 53:

Regarding claim 53, Jantz in view of Blumenau fails to teach or suggest wherein said request to identify devices attached to the storage network which are available to a host system is for devices available to the host system through a specified one of the ports, and wherein said requesting the storage network to identify devices is made for the specified port. The Examiner admits that Jantz does not teach or suggest requesting a storage network to identify devices available to the host system through a specified one of a plurality of ports of the host system to the fabric. The Examiner cites column 7, lines 1-10 of Blumenau and argues that Blumenau "teaches identifying the device and the port of the device that coupled to the network." However, claim 53 recites identifying devices attached to the storage network that are available to a host system through a specified port of the host system. Blumenau only teaches identifying each port of a device that couples to the network, but fails to teach anything regarding identifying those devices available through a specified port of the host system. Since neither Jantz nor Blumenau teaches identifying devices available through a specified port of the host system, the Examiner's proposed combination of Jantz and Blumenau would also fail to include identifying devices available through a specified port of the host system.

<u>Claim 54</u>:

Regarding claim 54, Jantz in view of Blumenau fails to teach or suggest wherein said request to identify devices attached to the storage network which are available to a host system is for devices available to the host system through a specified set of the ports, and wherein said requesting the storage network to identify devices is made for the specified set of the ports. The Examiner admits that Jantz does not teach or suggest requesting a storage network to identify devices available to the host system through a specified set of a plurality of ports of the host system to the fabric. The Examiner cites column 7, lines 1-10 of Blumenau and argues that Blumenau "teaches identifying the device and the port of the device that coupled to the network." However, claim 54 recites identifying devices attached to the storage network that are available to a host system through a specified set of ports *of the host system.* Blumenau only teaches identifying each port of a device that couples to the network, but fails to teach anything regarding identifying those devices available through a specified port of the host system. Since neither Jantz nor Blumenau teaches identifying devices available through a specified set of ports of the host system, the Examiner's proposed combination of Jantz and Blumenau would also fail to include identifying devices available through a specified set of ports of the host system.

Fourth Ground of Rejection:

Claims 10, 11, 53 and 54 stand finally rejected under 35 U.S.C. § 103(a) as being unpatentable over Jantz in view of Nolan and further in view of Blumenau et al. (U.S. Patent 6,665,714) (hereinafter "Blumenau"). Appellants traverse this rejection for at least the following reasons. Different groups of claims are addressed under their respective subheadings.

The Examiner rejects claims 10, 11, 53, and 54 under 35 U.S.C. § 103(a) as being unpatentable over Jantz in view of Blumenau (see above) and as being unpatentable over Jantz in view of Nolan in further view of Blumenau. Since the Examiner does not rely upon any portion of Nolan for his rejection of claims 10, 11, 53, or 54, the arguments above regarding the rejections of claims 10, 11, 53, or 54, over Jantz in view of Blumenau apply here as well.

Claim 10:

Regarding claim 10, as noted above, Jantz in view of Blumenau fails to teach or suggest wherein said request to identify devices attached to the storage network which are available to a host system is for devices available to the host system through a specified one of the ports, and wherein said requesting the storage network to identify devices is made for the specified port. The Examiner admits that Jantz does not teach or suggest requesting a storage network to identify devices available to the host system through a specified one of a plurality of ports of the host system to the fabric. The Examiner relies upon column 7, lines 1-10 of Blumenau and argues that Blumenau "teaches identifying the device and the port of the device that coupled to the network." However, claim 10 recites identifying devices attached to the storage network that are available to a host system through a specified port *of the host system*. Blumenau only teaches identifying each port of a device that couples to the network, but fails to teach anything regarding identifying those devices available through a specified port *of the host system*. The Examiner does not rely upon Nolan nor does Nolan overcome any of the deficiencies of Jantz and Blumenau. Since Jantz, Blumenau and Nolan all fail to teach or suggest identifying devices available through a specified port of the host system, the Examiner's proposed combination of Jantz, Blumenau and Nolan would also fail to include identifying devices available through a specified port of the host system.

Claim 11:

As noted above, Jantz in view of Blumenau fails to teach or suggest wherein said request to identify devices attached to the storage network which are available to a host system is for devices available to the host system through a specified set of the ports, and wherein said requesting the storage network to identify devices is made for the specified set of the ports. Nolan also fails to teach or suggest requesting the identifying of devices available to a host system through a specified set of ports and the Examiner does note rely upon Nolan in his rejection of claim 11. The Examiner also admits that Jantz does not teach or suggest requesting a storage network to identify devices available to the host system through a specified set of a plurality of ports of the host system to the fabric.

As with the rejection of claim 10, the Examiner cites column 7, lines 1-10 of Blumenau and argues that Blumenau "teaches identifying the device and the port of the device that coupled to the network." However, like claim 10, claim 11 recites identifying devices attached to the storage network that are available to a host system through a specified set of ports *of the host system*, while Blumenau only teaches identifying each port *of a device* that couples to the network. Blumenau fails to teach anything regarding identifying those devices available through a specified port of the host system. Since none of either Jantz, Blumenau, or Nolan teaches identifying devices available through a specified set of ports of the host system, the Examiner's proposed combination of Jantz, Blumenau and Nolan would also fail to include identifying devices available through a specified set of ports of the host system.

Claim 53:

The Examiner rejection claim 53 for the same reasons and for the same rational as the rejection of claim 10, described above. Therefore, the specific arguments regarding the rejection of claim 10 above apply to the rejection of claim 53 as well. Specifically, Jantz in view of Nolan in further view Blumenau fails to teach or suggest wherein said request to identify devices attached to the storage network which are available to a host system is for devices available to the host system through a specified one of the ports, and wherein said requesting the storage network to identify devices is made for the specified port. As noted above, since none of either Jantz, Nolan, or Blumenau teaches identifying devices available through a specified port of the host system, the Examiner's proposed combination of Jantz, Nolan, and Blumenau would also fail to include identifying devices available through a specified port of the host system.

Claim 54:

The Examiner rejection claim 54 for the same reasons and for the same rational as the rejection of claim 11, described above. Therefore, the specific arguments regarding the rejection of claim 11 above apply to the rejection of claim 54 as well. Specifically, Jantz in view of Nolan in further view of Blumenau fails to teach or suggest wherein said request to identify devices attached to the storage network which are available to a host system is for devices available to the host system through a specified set of the ports, and wherein said requesting the storage network to identify devices is made for the specified set of the ports. As noted above, since neither Jantz, Nolan, nor Blumenau teaches identifying devices available through a specified set of ports of the host system, the Examiner's proposed combination of Jantz, Nolan, and Blumenau would also fail to include identifying devices available through a specified set of ports of the host system.

Fifth Ground of Rejection:

Claims 13 and 56 stand finally rejected under 35 U.S.C. § 103(a) as being unpatentable over Jantz in view of Carlson and Basham. Appellants assert that claims 13 and 56 are patentable for at least the reasons presented above regarding their respective independent claims.

Sixth Ground of Rejection:

Claims 13 and 56 stand finally rejected under 35 U.S.C. § 103(a) as being unpatentable over Jantz in view of Nolan and further in view of Carlson et al. (U.S. Patent 5,600,791) (hereinafter "Carlson") and Basham et al. (U.S. Patent 6,182,167) (hereinafter "Basham"). Appellants assert that claims 13 and 56 are patentable for at least the reasons presented above regarding their respective independent claims.

Seventh Ground of Rejection:

Claims 15 and 58 stand finally rejected under 35 U.S.C. § 103(a) as being unpatentable over Jantz in view of Wieland (U.S. Patent 6,643,748). Appellants assert that claims 15 and 58 are patentable for at least the reasons presented above regarding their respective independent claims.

Eighth Ground of Rejection:

Claims 15 and 58 stand finally rejected under 35 U.S.C. § 103(a) as being unpatentable over Jantz in view of Nolan and further in view of Wieland. Appellants assert that claims 15 and 58 are patentable for at least the reasons presented above regarding their respective independent claims.

Ninth Ground of Rejection:

The Examiner rejected claims 23-25, 28-30 and 40-42 under 35 U.S.C. § 103(a) as being unpatentable over Blumenau in view of Nolan. Appellants traverse this rejection for at least the following reasons. Different groups of claims are addressed under their respective subheadings.

Claims 23, 24, 25, 28, 40 and 41:

Regarding claim 23, Blumenau in view of Nolan does not teach or suggest a fabric driver that is configured to online a selected subset of fabric devices so that the selected subset of fabric devices are accessible from the host system, wherein the fabric driver is further configured to create operating system device nodes within the host system for each device of the selected subset, wherein each operating system device node provides a mechanism for accessing a corresponding one of the subset of fabric devices through the operating system executing on the host system. Instead, Blumenau teaches a method for configuring storage systems to determine whether devices are authorized to access data according to the identity of the devices (Blumenau -- col. 1, line 42 – col. 2, line 12). Blumenau has nothing to do with creating operating system device nodes for accessing devices through a host system's operating system.

The Examiner refers to Blumenau's description at col. 23, line 60 – col. 24, line 25 of an interface for displaying information about devices. Appellants fail to see how this has any relevance to Appellants' claimed invention. Blumenau's teachings have nothing to do with creating operating system device nodes within a host system for each device of a selected subset. Blumenau pertains to the configuration of storage systems at the application level.

The Examiner admits that Blumenau does not teach wherein the fabric driver is further configured to online the selected subset of fabric devices so that the selected subset of fabric devices are accessible from the host system, wherein the fabric driver is further configured to create operating system device nodes within the host system for each device of the selected subset, wherein each operating system device node provides a mechanism for accessing a corresponding one of the subset of fabric devices through the operating system executing on the host system. The Examiner relies upon Nolan, citing FIGs 22-24 of Nolan and asserting, "Nolan teaches creating LUNs within the host system for each device and accessing LUNs through the operating system and selecting subset information within disks or LUNs."

However, none of the cited Figures having anything to do with a fabric driver as recited in Appellants claim 23. Instead, the cited figures of Nolan and the corresponding descriptions describe various screens of Nolan's user interface for configuring a storage server. For example, FIG. 22 illustrates a user interface including a hierarchical tree structure for displaying information regarding storage elements. Neither displaying and manipulating information regarding storage elements nor configuring storage elements of a storage server teach or suggest a fabric driver configured to online fabric devices so that the devices are accessible from a host system and that is further configured to create operating system device nodes for the devices. Thus, Blumenau and Nolan, both singly and in combination, clearly fail to teach or suggest the limitations of Appellants' claim 23.

Claim 29:

Regarding claim 29, Blumenau in view of Nolan fails to teach or suggest wherein the application is further configured to make said request to the fabric driver for a specified one of the one or more adapter ports of the host system. The Examiner cites column 7, lines 1-10 of Blumenau. However, as noted previously, this portion of Blumenau teaches identifying a device and the port through which the device is coupled to the network. Blumenau fails to mention requesting a fabric driver to provide a list of fabric devices for a specified adaptor port of the host system.

Jantz in view of Blumenau fails to teach or suggest wherein said request to identify devices attached to the storage network which are available to a host system is for devices available to the host system through a specified one of the ports, and wherein said requesting the storage network to identify devices is made for the specified port. The Examiner admits that Jantz does not teach or suggest requesting a storage network to identify devices available to the host system through a specified one of a plurality of ports of the host system to the fabric. The Examiner cites column 7, lines 1-10 of Blumenau and argues that Blumenau "teaches identifying the device and the port of the device that coupled to the network." However, claim 29 recites identifying devices attached to the storage network that are available to a host system through a specified port of the host system. Blumenau only teaches identifying each port of a device that couples to the network, but fails to teach anything regarding identifying those devices available through a specified port *of the host system.* Since neither Jantz nor Blumenau teaches identifying devices available through a specified port of the host system, the Examiner's proposed combination of Jantz and Blumenau would also fail to include identifying devices available through a specified port of the host system.

<u>Claim 30</u>:

Regarding claim 30, Blumenau in view of Nolan fails to teach or suggest wherein said request to identify devices attached to the storage network which are available to a host system is for devices available to the host system through a specified set of the ports, and wherein said requesting the storage network to identify devices is made for the specified set of the ports. The Examiner admits that Jantz does not teach or suggest requesting a storage network to identify devices available to the host system through a specified set of a plurality of ports of the host system to the fabric. The Examiner cites column 7, lines 1-10 of Blumenau and argues that Blumenau "teaches identifying the device and the port of the device that coupled to the network." However, claim 30 recites identifying devices attached to the storage network that are available to a host system through a specified set of ports *of the host system.* Blumenau only teaches identifying each port of a device that couples to the network, but fails to teach anything regarding identifying those devices available through a specified port of the host system. Since neither Jantz nor Blumenau teaches identifying devices available through a specified set of ports of the host system, the Examiner's proposed combination of Jantz and Blumenau would also fail to include identifying devices available through a specified set of ports of the host system.

<u>Claim 42</u>:

Regarding claim 42, Blumenau in view of Nolan fails to teach or suggest a Fibre Channel protocol module configured to perform SCSI protocol operations between the host system and the fabric; and one or more Fibre Channel port drivers configured to perform transport layer operations between the host system and the fabric wherein the Fibre Channel protocol module and the one or more Fibre Channel port drivers are part of an operating system kernel on the host system. The Examiner cites column 7, lines 1-35 of Blumenau. However, the cited passage fails to mention a Fibre Channel protocol module and one or more Fibre Channel port drivers that are part of an operating system kernel on the host system. Instead, the cited passage of Blumenau describes how the device of a Fibre Channel interconnect may communicate through various protocols. However, Blumenau is discussing how devices communicate amongst themselves and does not mention anything regarding a Fibre Channel protocol module or Fibre Channel port drivers that are a part of an operating system kernel on the host system. For example, Blumenau teaches, "devices coupled using the Fibre Channel network *may communicate with each other* using the same protocols that have historically been used" (emphasis added, Blumenau, column 7, lines 27-30).

As Nolan is not relied upon by the Examiner and since Nolan doesn't teach or suggest a Fibre Channel protocol module and one or more Fibre Channel port drivers that are part of an operating system kernel on the host system, Nolan fails to overcome any deficiencies of Blumenau regarding claim 42. Thus, the Examiner's proposed combination of Blumenau and Nolan fails to include a Fibre Channel protocol module and one or more Fibre Channel port drivers that are part of an operating system kernel on the host system.

Tenth Ground of Rejection:

Claim 26 stands finally rejected under 35 U.S.C. § 103(a) as being unpatentable over Blumenau in view of Nolan and further in view of Nolan (U.S. Patent 6,466,141) (hereinafter "Nolan 141"). Appellants traverse this rejection for at least the following reasons.

Claim 26:

Regarding claim 26, Blumenau in view of Nolan and further in view of Nolan141 fails to teach or suggest a fabric driver configured to receive the information about the fabric devices from the nameserver. The Examiner admits that Blumenau fails to teach a fabric driver that receives information about fabric devices from a nameserver, but cites column 26, lines 1-5 of Nolan141. The Examiner argues that Nolan141 teaches "receiving requests for storage transactions on the server interface, to direct the requested storage transactions to the plurality of storage devices." Presumably the Examiner is equating receiving and directing storage transaction requests to receiving information about fabric devices from a nameserver. However, receiving and directing storage transaction requests is not the same as, nor has anything to do with receiving information about fabric devices from a nameserver. Thus, the cited passage of Nolan141 has no relevance to a fabric driver configured to receive the information about the fabric devices from the nameserver. The Examiner does not rely upon Nolan, nor does Nolan overcome any deficiencies of either Blumenau or Nolan141 regarding a fabric driver configured to receive the information about the fabric devices from the nameserver.

Further regarding claim 26, Blumenau in view of Nolan and further in view of Nolan141 fails to teach or suggest selecting the fabric devices supporting one protocol out of a plurality of protocols supported on the fabric, from the information about the fabric devices received from the nameserver. The Examiner cites column 24, lines 10-67 of Blumenau. The cited passage refers to a user interface allowing users to view information regarding devices, such as which host processors can communicate with a storage system, the path or physical connection by which they communicate, which storage volumes of data accessible to a particular host processes, etc. (Blumenau, column 14, lines 4-8). However, Blumenau fails to mention, at the cited passage or elsewhere, selecting fabric devices supporting *one particular protocol out of a plurality of protocols* supported by the fabric. Furthermore, as neither Nolan nor Nolan141 teaches or suggests anything about selecting fabric devices supporting one particular protocol out of a plurality of protocols supported by the fabric, the Examiner's proposed combination of

Blumenau, Nolan and Nolan 141, also fails to include selecting fabric devices supporting one particular protocol out of a plurality of protocols supported by the fabric.

Blumenau in view of Nolan and further in view of Nolan141 also fails to teach or suggest wherein the fabric driver is configured to return the list of fabric devices to the application, wherein the list of fabric devices is a list of devices supporting the one protocol. The Examiner relies upon column 24, lines 10-67 of Blumenau. However, as noted above, the cited passage refers to a user interface allowing users to view information regarding devices, such as which host processors can communicate with a storage system, the path or physical connection by which they communicate, which storage volumes of data accessible to a particular host processes, etc. The cited passage does not teach or suggest anything regarding returning the list of fabric devices supporting the one particular protocol to the application. Additionally, since neither Nolan nor Nolan141 are relied upon by the Examiner, and since neither Nolan nor Nolan141 teaches or suggests returning to the application a list of fabric devices supporting the one particular protocol, the Examiner's proposed combination of Blumenau, Nolan and Nolan141 also fails to teach or suggest a fabric driver configured to return the list of fabric devices to the application, wherein the list of fabric devices is a list of devices supporting the one protocol.

Eleventh Ground of Rejection:

Claim 27 stands finally rejected under 35 U.S.C. § 103(a) as being unpatentable over Blumenau in view of Nolan and further in view of Nolan '141 and Sambamurthy et al. (U.S. Patent 6,393,489) (hereinafter "Sambamurthy"). Claim 27 is patentable for at the reasons presented above regarding its respective independent claim.

Twelfth Ground of Rejection:

Claim 31 stands finally rejected under 35 U.S.C. § 103(a) as being unpatentable over Blumenau in view of Nolan and further in view of Jantz. As claim 31 was canceled in the Response to Office Action dated July 15, 2004, the rejection of claim 31 is moot.

<u>Thirteenth Ground of Rejection:</u>

Claims 32-37 stand finally rejected under 35 U.S.C. § 103(a) as being unpatentable over Blumenau in view of Nolan and further in view of Chow et al. (U.S. Patent 6,594,698) (hereinafter "Chow"). Appellants traverse this rejection for at least the following reasons. Different groups of claims are addressed under their respective subheadings.

<u>Claims 32, 33, 34 and 35</u>:

Regarding claim 32, Blumenau in view of Nolan in further view of Chow fails to teach or suggest a device discovery mechanism configured to <u>determine whether each of the I/O ports is connected to one or more direct attach devices or to the fabric</u>. The Examiner admits that Blumenau fails to teach the limitations of claim 32 and relies upon Chow, citing column 15, lines 15-55. However, the cited portion of Chow does not teach or suggest determining whether each of the I/O ports is connected to one or more direct attach devices or to the fabric. Instead, the cited passage describes how Chow's I/O nodes or IONs work together to determine a globally unique identification number for each ION. For example, Chow teaches, "[o]ne technique of obtaining a globally unique ION identifier 604 is to use the electronically readable motherboard serial number that is often stored in the real time clock chip" (Chow, column 15, lines 44-47). However, nowhere does Chow mention anything regarding a device discovery mechanism configured to determine whether each of the I/O ports is connected to one or more direct attach devices or to the fabric. As Nolan also fails to teach or suggest a device discovery mechanism configured to determine whether each of the I/O ports is connected to one or more direct attach devices or to the fabric, the Examiner's proposed combination of Blumenau, Nolan and Chow also fails to teach or suggest anything regarding determining whether each of the I/O ports is connected to one or more direct attach devices or to the fabric.

Additionally, Blumenau in view of Nolan in further view of Chow further fails to teach or suggest a device discovery mechanism configured to, for each of the I/O ports connected to one or more direct attach devices, discover the one or more direct attach devices and create an operating system node for accessing each direct attach device. The Examiner relies upon column 15, lines 15-55 of Chow and argues that Chow teaches, "the compute nodes 200 create an entry point to each fabric virtual disk at boot time and update those entry points dynamically using a naming protocol established between the compute nodes 200 and the IONs 212." The Examiner is presumably referring to column 40, lines 50-55, where Chow includes that statement. However, Chow is describing how multiple IONs may partition virtual and physical storage disks among them and create a list of which virtual disks are owned by which ION (chow, column 40, lines 30-39). However, Chow is teaching the partitioning of fabric devices and the creation of virtual disks for those fabric devices and is not teaching anything about creating operating system nodes for accessing direct attach devices, contrary to the Examiner's assertion.

Blumenau in view of Nolan in further view of Chow also fails to teach or suggest for each of the I/O ports connected to the fabric, designate the I/O port as a fabric port without attempting to discover the fabric devices. As noted above, the Examiner's cited passage of Chow describes how Chow's I/O nodes or IONs work together to determine a globally unique identification number for each ION. Also as noted above, the cited passage does not mention determining whether an I/O port is connected to a direct attach device or to the fabric. Furthermore, the cited passage is also silent regarding designating an I/O port connected to the fabric as a fabric port without attempting to discover the fabric devices. Thus, the Examiner's cited passage does not make any reference to designating I/O ports as fabric ports without attempting to discover the fabric devices.

As Nolan also fails to teach or suggest the limitations of claim 32, the Examiner proposed combination of Blumenau, Nolan and Chow also fails to include determining whether each of the I/O ports is connected to one or more direct attach devices or to the fabric; for each of the I/O ports connected to one or more direct attach devices, discover the one or more direct attach devices and create an operating system node for accessing each direct attach device; and for each of the I/O ports connected to the fabric, designate the I/O port as a fabric port without attempting to discover the fabric devices.

Furthermore the Examiner has not provided a proper motivation to combine the teachings of Chow with those of Blumenau and Nolan. The Examiner merely states that it would have been obvious to combine Chow's teachings "to transmit data among devices via [a] network system." However, merely transmitting data among devices via a network system does not provide any motivation to combine Chow with Blumenau and Nolan as each of their respective teachings already provide for transmitting data among devices. Thus, no one seeking to transmit data among devices via a network system would not be motivated to combine the respective teachings of Blumenau, Nolan and Chow, as suggested by the Examiner.

<u>Claim 36</u>:

Blumenau in view of Nolan in further view of Chow fails to teach or suggest wherein the discovery mechanism is configured to determine whether each of the I/O ports is connected to one or more direct attach devices or to the fabric <u>by attempting to log-in to the fabric through each I/O port</u>. The Examiner cites FIGs. 11 and 16 of Blumenau without providing any interpretation or argument regarding how either FIG. 11 or FIG 16 teaches this limitation of claim 36. In fact, neither FIGs. 11 and 16, nor any other portion of Blumenau describes a discovery mechanism configured to determine whether each of the I/O ports is connected to one or more direct attach devices or to the fabric by attempting to log-in to the fabric through each I/O port.

Instead, according to Blumenau, "FIG. 11 is a flow diagram illustrating one method of validating a data transfer between an initiator and target using the checksum of FIG. 10 according to one embodiment of the invention" (Blumenau, column 4, lines 23-26). Blumenau describes FIG. 16 as illustrating "one method of graphically representing how data is stored on a storage system that can be provided by a graphical user interface according to one embodiment of the present invention" (Blumenau, column 4, lines 42-45). Clearly neither FIG. 11 nor FIG. 16, nor their respective accompanying text, has anything to do with determining whether an I/O port is connected to direct attached devices or the fabric by attempting to log-in to the fabric through the I/O port.

Furthermore, Blumenau in view of Nolan in further view of Chow also fails to teach or suggest wherein if the log-in fails, said discovery mechanism is configured to designate the I/O port as a direct-attach port; and if the log-in is successful, designate the I/O port as a fabric port. As noted above, the Examiner merely cites FIGs. 11 and 16 of Blumenau, which, as described above, have nothing to do with designating an I/O port as either a direct-attach port or a fabric port depending upon whether an attempted log-in to the fabric through the port was successful.

As neither Nolan nor Chow is relied upon by the Examiner and since neither Nolan or Chow overcomes any of the above noted deficiencies of Blumenau regarding the limitations of claim 36, the Examiner's proposed combination of Blumenau, Nolan and Chow clearly fails teach or suggest wherein the discovery mechanism is configured to determine whether each of the I/O ports is connected to one or more direct attach devices or to the fabric by attempting to log-in to the fabric through each I/O port; wherein if the log-in fails, said discovery mechanism is configured to designate the I/O port as a direct-attach port; and if the log-in is successful, designate the I/O port as a fabric port.

Claim 37:

Blumenau in view of Nolan in further view of Chow fails to teach or suggest a library configured to provide an interface between the application and the fabric driver, wherein the library is configured to update a persistent repository for each fabric device successfully brought online for the host system to indicate which devices are currently online. The Examiner merely cites FIGs 18-21 of Blumenau without providing any interpretation or explanation regarding how FIGs. 18-21 relate to the limitations of claim 37. FIGs. 18 – 21 of Blumenau illustrate various steps in configuring access to a storage system from different hosts in a storage network, none of FIGs 18-21 nor any other portion of Blumenau describes a library configured to provide an interface between an application and a fabric driver.

Furthermore, Blumenau is silent about such a library updating a persistent repository to indicate which devices are currently online. Instead, Blumenau teaches that the user interface, which the Examiner equates to the application of claim 37, directly communicates and updates "a configuration database in the storage system to identify host processors that are logged into the storage system over the network, to identify storage volumes on the storage system, to identify whether access to a particular storage volume on the storage system is permitted from a particular host processor, and to identify a network path by which host processors are logged into the storage system over the network" (Blumenau, summary; column 9, lines 14-59; column 18, line 62-column 19, line 9). Thus, **Blumenau actually teaches away** from a library providing an interface between the application and the fabric driver that also updates a persistent repository to indicate which devices are currently online.

The Examiner does not rely upon Nolan or Chow in his rejection of claim 37 and neither Nolan nor Chow overcomes any of the above noted deficiencies of Blumenau regarding the limitations of claim 37.

Fourteenth Ground of Rejection:

Claims 38-39 stand finally rejected under 35 U.S.C. § 103(a) as being unpatentable over Blumenau in view of Nolan and further in view of Chow, Carlson and Basham. Claims 38 and 39 are patentable for at least the reasons presented above regarding their respective independent claims.

VIII. CONCLUSION

For the foregoing reasons, it is submitted that the Examiner's rejection of claims 1-15, 23-59, 67 and 68 was erroneous, and reversal of his decision is respectfully requested.

The Commissioner is authorized to charge the appeal brief fee of $500.00 and any other fees that may be due to Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C. Deposit Account No. 501505/5181-79200/RCK. This Appeal Brief is submitted with a return receipt postcard.

Respectfully submitted,

Robert C. Kowert
Reg. No. 39,255
Attorney for Appellants

Meyertons, Hood, Kivlin, Kowert & Goetzel, P.C.
P.O. Box 398
Austin, TX 78767-0398
(512) 853-8850

Date: June 10, 2005

IX. CLAIMS APPENDIX

The claims on appeal are as follows.

1. A method for discovering fabric devices, comprising:

receiving a list from a fabric driver of fabric devices available to a host system, wherein the fabric driver is part of an operating system for the host system;

receiving a request to select a subset of the fabric devices from the list; and

requesting the fabric driver to create an operating system device node in the host system for each of the fabric devices in the subset not already online, wherein each operating system device node provides a mechanism for accessing a corresponding one of the subset of fabric devices through the operating system executing on the host system.

2. The method as recited in claim 1, further comprising, prior to said receiving a request to select a subset of the fabric devices from the list:

displaying the list of fabric devices available to the host system.

3. The method as recited in claim 1, further comprising, prior to said receiving a list:

requesting the fabric driver to provide the list of fabric devices available to the host system in response to user input.

4. A method for discovering fabric devices, comprising:

providing a list of fabric devices available to a host system;

receiving a request to create operating system device nodes in the host system for each fabric device in a selected subset of the fabric devices available to the host system; and

creating an operating system device node in the host system for each of the fabric devices in the selected subset not already online, wherein each operating system device node provides a mechanism for accessing a corresponding one of the subset of fabric devices through an operating system executing on the host system.

5. The method as recited in claim 4, further comprising, prior to said providing a list of fabric devices:

querying a fabric nameserver for information about the fabric devices;

receiving the information about the fabric devices from the nameserver; and

compiling the list of fabric devices available to the host system.

6. The method as recited in claim 5, wherein said compiling the list comprises:

from the information about the fabric devices, selecting the fabric devices supporting one protocol out of a plurality of protocols supported on the fabric; and

compiling the list of fabric devices to list only those fabric devices supporting said one protocol.

7. The method as recited in claim 6, wherein said one protocol is SCSI over Fibre Channel.

8. The method as recited in claim 4, wherein the list comprises address information to address the fabric devices through the fabric.

9. A method for discovering devices attached to a storage network, comprising:

receiving a request to identify devices attached to the storage network which are available to a host system;

requesting the storage network to identify devices attached to the storage network which are available to the host system;

receiving a list of the identified devices;

receiving a request to on-line a subset of the identified devices; and

creating an operating system device node within the host system for each of the identified devices in the subset that is not already online, wherein each operating system device node provides a mechanism for accessing a corresponding one of the subset of the identified devices through an operating system executing on the host system.

10. The method as recited in claim 9, wherein the storage network comprises a fabric, and wherein the host system comprises a plurality of ports to the fabric, wherein said request to identify devices attached to the storage network which are available to a host system is for devices available to the host system through a specified one of the ports, and wherein said requesting the storage network to identify devices is made for the specified port.

11. The method as recited in claim 9, wherein the storage network comprises a fabric, and wherein the host system comprises a plurality of ports to the fabric, wherein said request to identify devices attached to the storage network which are available to a host system is for devices available to the host system through a specified set of the ports, and wherein said requesting the storage network to identify devices is made for the specified set of the ports.

12. The method as recited in claim 9, further comprising, for each device successfully brought online for the host system by said creating an operating system device node, updating a persistent repository to indicate which devices are currently online.

13. The method as recited in claim 12, further comprising:

receiving from the storage network a notification that a device is no longer available; and

updating the persistent repository to reflect that the unavailable device is offline.

14. The method as recited in claim 12, further comprising:

in response to a reboot of the host system:

reading the persistent repository; and

onlining the devices indicated by the persistent repository to have been onlined prior to the reboot.

15. The method as recited in claim 9, wherein the storage network comprises a Fibre Channel switched fabric comprising a plurality of Fibre Channel switches.

23. A host system, comprising:

one or more adapter ports for connecting to a fabric;

a fabric driver configured to interface the host system to the fabric, wherein the fabric driver is part of an operating system for the host system;

an application configured to request the fabric driver to provide a list of fabric devices attached to the fabric that are visible to the host system through one of said adapter ports;

wherein the fabric driver is further configured to provide the list of fabric devices to the application in response to the request from the application;

wherein the application is further configured to indicate to the fabric driver a selected subset of the fabric devices from the list to be brought online for access from the host system; and

wherein the fabric driver is further configured to online the selected subset of fabric devices so that the selected subset of fabric devices are accessible from the host system, wherein the fabric driver is further configured to create operating system device nodes within the host system for each device of the selected subset, wherein each operating system device node provides a mechanism for accessing a corresponding one of the subset of fabric devices through the operating system executing on the host system.

24. The host system as recited in claim 23, wherein the application is further configured to:

display the list to a user through a user interface; and provide through the user interface for the user to select devices from the list as the selected subset of the fabric device to be brought online.

25. The host system as recited in claim 23, wherein, in response to the request from the application, the fabric driver is further configured to:

query a fabric nameserver for information about the fabric devices to compile the list;

wherein the nameserver maintains information identifying devices accessible throughout the fabric.

26. The host system as recited in claim 25, wherein the fabric driver is further configured to:

receive the information about the fabric devices from the nameserver;

from the information about the fabric devices, select the fabric devices supporting one protocol out of a plurality of protocols supported on the fabric; and

return the list of fabric devices to the application, wherein the list of fabric devices is a list of devices supporting said one protocol.

27. The host system as recited in claim 26, wherein said one protocol is SCSI over Fibre Cannel.

28. The host system as recited in claim 23, wherein the list comprises address information to address the fabric devices through the fabric.

29. The host system as recited in claim 23, wherein the application is further configured to make said request to the fabric driver for a specified one of the one or more adapter ports.

30. The host system as recited in claim 23, wherein the application is further configured to make said request to the fabric driver for a specified set of the one or more adapter ports.

32. The host system as recited in claim 23, further comprising:

a plurality of I/O ports including the one or more adapter ports for connecting to a fabric; and

a device discovery mechanism configured to:

determine whether each of the I/O ports is connected to one or more direct attach devices or to the fabric;

for each of the I/O ports connected to one or more direct attach devices, discover the one or more direct attach devices and create an operating system node for accessing each direct attach device; and

for each of the I/O ports connected to the fabric, designate the I/O port as a fabric port without attempting to discover the fabric devices.

33. The host system as recited in claim 32, wherein said discovery mechanism is configured to execute in response to a reboot of the host system, and wherein said application is configured to execute on the host system subsequent to said reboot and said discovery process.

34. The host system as recited in claim 32, wherein each of the I/O ports connected to the fabric comprises a Fibre Channel host adapter port.

35. The host system as recited in claim 32, wherein each of the I/O ports connected to one or more direct attach devices comprises a port to a Fibre Channel private loop or point-to-point link.

36. The host system as recited in claim 32, wherein:

said discovery mechanism is configured to determine whether each of the I/O ports is connected to one or more direct attach devices or to the fabric by attempting to log-in to the fabric through each I/O port;

wherein if the log-in fails, said discovery mechanism is configured to designate the I/O port as a direct-attach port; and

if the log-in is successful, designate the I/O port as a fabric port.

37. The host system as recited in claim 32, further comprising a library configured to provide an interface between said application and said fabric driver, wherein the library is configured to update a persistent repository for each fabric device successfully brought online for the host system to indicate which devices are currently online.

38. The host system as recited in claim 37, wherein the library is further configured to:

receive from the fabric driver a notification that a fabric device is no longer available; and update the persistent repository to reflect that the unavailable fabric device is offline.

39. The host system as recited in claim 37, wherein the discovery mechanism is further configured to, in response to a reboot of the host system:

read the persistent repository; and

request the fabric driver to online the devices indicated by the persistent repository to have been onlined prior to the reboot.

40. The host system as recited in claim 23, wherein the fabric comprises a Fibre Channel switched fabric comprising a plurality of Fibre Channel switches.

41. The host system as recited in claim 23, wherein the fabric is part of a storage area network (SAN), and wherein the fabric devices comprise storage devices.

42. The host system as recited in claim 23, wherein the fabric driver comprises:

a Fibre Channel protocol module configured to perform SCSI protocol operations between the host system and the fabric; and

one or more Fibre Channel port drivers configured to perform transport layer operations between the host system and the fabric;

wherein the Fibre Channel protocol module and the one or more Fibre Channel port drivers are part of an operating system kernel on the host system.

43. A computer readable medium having stored thereon data representing

sequences of instructions, wherein the sequence of instructions are executable by one or more processors to implement:

receiving a list from a fabric driver of fabric devices available to a host system, wherein the fabric driver is part of an operating system for the host system;

receiving a request to select a subset of the fabric devices from the list; and

requesting the fabric driver to create an operating system device node in the host system for each of the fabric devices in the subset, wherein each operating system device node provides a mechanism for accessing a corresponding one of the subset of fabric devices through the operating system executing on the host system.

44. The computer readable medium as recited in claim 43, wherein said receiving a list, said selecting a subset, and said requesting the fabric driver to online the selected subset, are performed through an application executing on the host system.

45. The computer readable medium as recited in claim 43, wherein the program instructions are further configured to implement, prior to said receiving a request to select a subset of the fabric devices from the list:

displaying the list of fabric devices available to the host system.

46. The computer readable medium as recited in claim 43, wherein the program instructions are further configured to implement, prior to said receiving a list:

requesting the fabric driver to provide the list of fabric devices available to the host system in response to user input.

47. A computer readable medium having stored thereon data representing sequences of instructions, wherein the sequence of instructions are executable by one or more processors to implement:

providing a list of fabric devices available to a host system;

receiving a request to create operating system device nodes in the host system for each fabric device in a selected subset of the fabric devices available to the host system; and

creating an operating system device node in the host system for each of the fabric devices in the selected subset, wherein each operating system device node provides a mechanism for accessing a corresponding one of the subset of fabric devices through an operating system executing on the host system.

48. The computer readable medium as recited in claim 47, wherein the program instructions are further configured to implement, prior to said providing a list of fabric devices:

querying a fabric nameserver for information about the fabric devices;

receiving the information about the fabric devices from the nameserver; and

compiling the list of fabric devices available to the host system.

49. The computer readable medium as recited in claim 48, wherein said compiling the list comprises:

from the information about the fabric devices, selecting the fabric devices supporting one protocol out of a plurality of protocols supported on the fabric; and compiling the list of fabric devices to list only those fabric devices supporting said one protocol.

50. The computer readable medium as recited in claim 49, wherein said one protocol is SCSI over Fibre Channel.

51. The computer readable medium as recited in claim 47, wherein the list comprises address information to address the fabric devices through the fabric.

52. A computer readable medium having stored thereon data representing sequences of instructions, wherein the sequence of instructions are executable by one or more processors to implement:

receiving a request to identify devices attached to the storage network which are available to a host system;

requesting the storage network to identify devices attached to the storage network which are available to the host system;

receiving a list of the identified devices;

receiving a request to on-line a subset of the identified devices; and

creating a node within the host system for each of the identified devices in the subset that is not already online, wherein each operating system device node provides a mechanism for accessing a corresponding one of the subset of the identified devices through an operating system executing on the host system.

53. The computer readable medium as recited in claim 52, wherein the storage network comprises a fabric, and wherein the host system comprises a plurality of ports to the fabric, wherein said request to identify devices attached to the storage network which are available to a host system is for devices available to the host system through a specified one of the ports, and wherein said requesting the storage network to identify devices is made for the specified port.

54. The computer readable medium as recited in claim 52, wherein the storage network comprises a fabric, and wherein the host system comprises a plurality of ports to the fabric, wherein said request to identify devices attached to the storage network which are available to a host system is for devices available to the host system through a specified set of the ports, and wherein said requesting the storage network to identify devices is made for the specified set of the ports.

55. The computer readable medium as recited in claim 52, wherein the program instructions are further configured to implement, for each device successfully brought online for the host system by said creating an operating system device node, updating a persistent repository to indicate which devices are currently online.

56. The computer readable medium as recited in claim 55, wherein the program instructions are further configured to implement:

receiving from the storage network a notification that a device is no longer available; and

updating the persistent repository to reflect that the unavailable device is offline.

57. The computer readable medium as recited in claim 55, wherein the program instructions are further configured to implement:

in response to a reboot of the host system:

reading the persistent repository; and

onlining the devices indicated by the persistent repository to have been onlined prior to the reboot.

58. The computer readable medium as recited in claim 52, wherein the storage network comprises a Fibre Channel switched fabric comprising a plurality of Fibre Channel switches.

59. The computer readable medium as recited in claim 52, wherein the storage network is part of a storage area network (SAN), and wherein the devices comprise storage devices.

67. A method for discovering fabric devices, comprising:

viewing a list of fabric devices available to a host system;

selecting a subset of the fabric devices from the list; and

requesting that each of the fabric devices in the subset be brought online if not already online for use from the host system, wherein each fabric device that is online has a corresponding operating system device node that provides a mechanism for accessing a corresponding one of the subset of the identified devices through an operating system executing on the host system.

68. The method as recited in claim 67, further comprising, prior to said viewing a list:

requesting the list of fabric devices available to the host system.

X. EVIDENCE APPENDIX

No evidence submitted under 37 CFR §§ 1.130, 1.131 or 1.132 or otherwise entered by the Examiner is relied upon in this appeal.

XI. RELATED PROCEEDINGS APPENDIX

There are no related proceedings.